

Improved treebank querying: a facelift for GrETEL

Liesbeth Augustinus, Bram Vanroy and Vincent Vandeghinste

Centre for Computational Linguistics

University of Leuven, Belgium

{liesbeth, vanroy, vincent}@ccl.kuleuven.be

Abstract

We describe the improvements to the interface of GrETEL, an online tool for querying treebanks. We demonstrate how two usability tests and individual user feedback resulted in a more user-friendly interface which meets the users' needs.

1 Introduction

It has become common practice for linguists to use text corpora, mainly for the purpose of retrieving examples and obtaining quantitative data about certain phenomena. For syntactic research, syntactically annotated corpora or *treebanks* are of special interest. Descriptive linguists are often using corpus data to support their theoretical claims, but are often reluctant towards using treebanks. This is mainly due to the limited user-friendliness of the search tools and the lack of standardization in grammatical frameworks, formalisms, and query languages. This is unfortunate, since many types of grammatical phenomena, such as discontinuous constructions, are extremely hard to find in raw text or POS-tagged corpora, but can be found rather easily in treebanks. In order to make treebanks accessible to users with a limited computational background, GrETEL (Greedy Extraction of Trees for Empirical Linguistics) was developed as a user-friendly search engine for treebanks (Augustinus et al., 2012; Augustinus et al., in press), and conforms to the CLARIN goal to open up language resources for human and social sciences.

2 How GrETEL works

The general architecture of GrETEL has remained unchanged. We provide two ways to query a treebank: via a natural language example or via an XPath query.¹ Linguists often use example sentences as a starting point for their research. GrETEL allows them to use those examples as a starting point for the treebank investigation. The first query mode is the example-based querying mode. First, the user provides an example containing the phenomenon under investigation. Second, the input is automatically syntactically analysed with the Alpino parser (van Noord, 2006). Third, the user indicates which parts of the example are important for the search, which is automatically converted into an XPath expression. This expression may be manually adapted by the user. Fourth, the user selects the treebank(s) that should be queried. Fifth, an overview of the search instruction is given. Sixth, the actual search is performed and the results (if any) are presented to the user. The second query mode is entering an XPath query straight away. This allows more flexible queries, but requires mastering the formal query language. For a more detailed description of the GrETEL work flow, cf the examples in Augustinus et al. (2012) and Augustinus et al. (in press).

3 Improving the GrETEL interface

In order to ensure user-friendliness of the query engine we hired a company specialised in User Centred Design (UCD) to set up two user tests, resulting in a number of suggestions for interface improvements (section 3.1), which led to GrETEL 2.0 (section 3.2). A second series of user tests showed that many issues were solved, but some issues remained. A number of them were addressed in the third version of GrETEL (section 3.3).

This work is licenced under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>

¹<https://www.w3.org/TR/xpath>

3.1 User interface testing

After the release of GrETEL 1.0,² we asked five linguists from different genders and ages to test the system.³ They were asked to execute nine scenarios (look up specific syntactic phenomena in the treebank) within 45 minutes, while being observed. An example scenario is given in (1).

- (1) Dutch has some constructions where the auxiliary of the perfect tense can appear as a form of either *hebben* ‘have’ or *zijn* ‘be’, e.g. *Hij heeft/is kunnen komen* ‘He was able to come’ (lit. ‘He has/is can come’).

Use GrETEL to investigate whether you can find instances of this kind of variation.

The users did not get any training on the use of GrETEL. A manual (text in PDF-format) was available on the website, but no one took the effort to consult it. The users were asked to think aloud and comment on what they were doing and why they were doing it. The developers of GrETEL observed the users from a different room via screen sharing and audiovisual recording, which we experienced as a frustrating but eye-opening experience. Some users did not complete any scenario, and the best performing users could only complete three scenarios. This was often due to the fact that users had misconceptions or wrong expectations about the query tool. Nevertheless, the tests clearly showed the need for improvement in the interface. Figure 1 presents a screenshot of GrETEL 1.0.

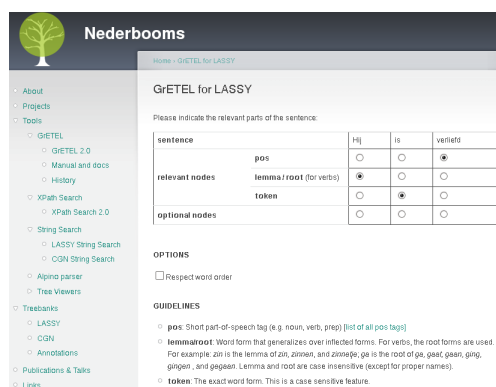


Figure 1: Screenshot of the GrETEL 1.0 interface

The main issue was that it was unclear that the query process consists of several steps (as explained in section 2). There was no indication of progress in the query procedure, and it was unclear that the users were still building the query while they were already expecting results. In general there was a lack of clearly formulated, concise instructions.

A second issue was that users understood the instruction ‘Please provide an input example’ in a different way as was meant by the developers. Some users wanted to insert part-of-speech tags in their query, as the instructions did not clearly indicate that the tool requires an example sentence as input.

A third issue was the fact that there was a left-hand side navigation menu, which contained links to the research project, and to many other things that were not strictly part of the query process. The users could not make a proper distinction between what was actually part of the query they were executing, and what was part of the general project website.

A number of smaller issues were, amongst others, that some search options were unclear, that there was no help available through pop-ups or mouse hovering, that some buttons and results appeared under the fold, i.e. they were not visible to the user without scrolling and the users were unaware that buttons

²The first version of GrETEL is still available via <http://nederbooms.ccl.kuleuven.be/eng/history>.

³Five test persons may seem few, but the UCD company suggested using a small test group for qualitative, in-depth usability studies. As we also receive feedback from users outside this study, this is not a major problem. The restriction to linguists is due to the main goal of the first version of GrETEL, i.e. make treebanks accessible for descriptive linguists. Other HSS researchers could also benefit from GrETEL, e.g. historians or literature researchers who want to query annotated historical or literary documents, but currently no such documents are included in GrETEL. Future versions might contain such treebanks.

or results were shown on the bottom of the page, that buttons were not consistently positioned in the different steps and that all users were confronted with an adaptable XPath expression.

3.2 From GrETEL 1.0 to GrETEL 2.0

Each of the above mentioned issues were addressed in GrETEL 2.0.⁴ We clearly marked the different steps in the query process, indicating the current step, and allowing the user to navigate between the steps, cf the screenshot in Figure 2. We formulated the instructions with more care and detail, and provided help using hovering tooltips. The left-side navigation pane was removed. We changed the results screen such that users no longer need to scroll down in order to see the first results.

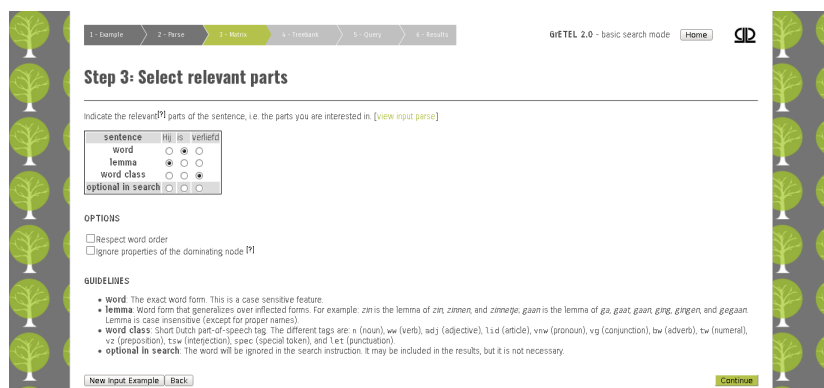


Figure 2: Screenshot of the GrETEL 2.0 interface

After addressing these issues in the GrETEL 2.0 interface, a second iteration of the user test was performed, with another five linguists. It was clear that the user interface had improved. Users now managed to complete most scenarios and obtain the results. They still had a number of remarks, of which we implemented the easy ones. For instance, we included a ‘loading image’ which is shown while the actual treebank search is performed. In addition to the update of the interface, we also added a tutorial and some additional exercises to the GrETEL website. Some other remarks remained on our to-do list while a last set of remarks is outside the scope of what we intend with GrETEL, such as adding tools for spelling correction.

3.3 From GrETEL 2.0 to GrETEL 3

After this series of usability tests, we received many positive reactions from users with respect to the increased user-friendliness of GrETEL. We have implemented most of the remaining remarks resulting from the second user test and individual user feedback GrETEL 3.⁵ The major changes with respect to GrETEL 2.0 include the interface design, the visualisation of the syntax trees, and search algorithm improvements.

Design Instead of the full-fluid design, the width of the website is restricted to 1024px which can scale downward. This causes the website to look less empty, and puts the focus on the centre of the screen. The different steps of the query procedure in the navigation bar are now clickable, making it easier for the user to go back in the query process and adapt the search instruction. A top menu was added to facilitate general navigation on the website. In light of making everything clear to the user, but without cluttering the website with distracting content, there is a separate documentation page with manuals, reference links, tutorials, and a Frequently Asked Questions section.

Query builder and tree visualisation Another remarkable change is the dynamic query builder, cf the screenshot in Figure 3. When the user selects the relevant parts of the example in the matrix, the shown query tree is adapted on the spot. This makes query building more intuitive as the user immediately sees

⁴<http://gretel.ccl.kuleuven.be/gretel-2.0>

⁵<http://gretel.ccl.kuleuven.be/gretel3>

the implications of (de)selecting items in the matrix. Moreover, we got rid of the choice between basic and advanced search in step one, and added a button to show or hide the advanced search options to the query building step.

The visualization of the syntax trees is updated from an SVG to a HTML and CSS approach. The visualizer is compatible with all modern browsers, and allows full JavaScript integration. The user can now click on the tree nodes and inspect all the linguistic information that the tree includes. The tree structure is shown using the full page width and it is possible to zoom in to inspect large trees.

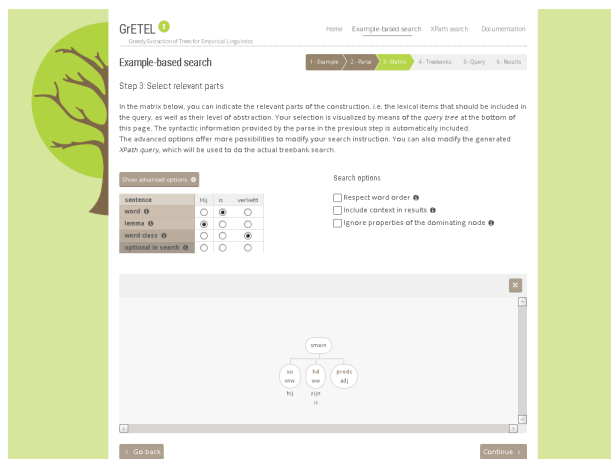


Figure 3: Screenshot of the GrETEL 3 interface

Search algorithm improvements The most important change to the server-side is that search results are shown as soon as they are found, whereas in previous versions the user has to wait until all results are gathered. Instead of looking for all results at once, GrETEL sends a lot of sequential requests to the server to ask for a single result: the next match. This implies that as soon as the first hit is found, it is presented to the user. In parallel, another general request is sent asking the server for all results at once. This process will generally take a lot of time to finish, but in some cases it finishes quicker than looking for all individual hits. If one process finishes (i.e. if all results are found), the other terminates as well. So, the best of two worlds is combined: all results are found as fast as possible (either by returning one result at a time, or all at once), and the user does not have to wait until all results are gathered.

4 Conclusion and future work

We presented how the GrETEL interface is optimized based on the results of two usability studies and individual user feedback. In future work we will integrate the results of a related project which allows the filtering of metadata in the search results, and the addition of options to intelligently sort the search results. Furthermore, we will keep on collecting user feedback and suggestions to improve the tool.

References

- Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde. 2012. Example-Based Treebank Querying. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3161–3167, Istanbul.
- Liesbeth Augustinus, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde, in press. *GrETEL: A tool for example-based treebank mining*, chapter 22. Ubiquity Press, London.
- Gertjan van Noord. 2006. At Last Parsing Is Now Operational. In *Proceedings of TALN*, pages 20–42.